An easy-to-use evaluation framework for benchmarking entity recognition and disambiguation systems

Chen, Hui; Wei, Bao-gang; Li, Yi-ming; Liu, Yonghuai; Zhu, Wen-hao

Published in:
Frontiers of Information Technology & Electronic Engineering

DOI:
10.1631/FITEE.1500473

Publication date:
2017

Citation for published version (APA):
Chen, H., Wei, B., Li, Y., Liu, Y., & Zhu, W. (2017). An easy-to-use evaluation framework for benchmarking entity recognition and disambiguation systems. *Frontiers of Information Technology & Electronic Engineering*, *18*(2), 195-205. https://doi.org/10.1631/FITEE.1500473

An Easy-to-Use Evaluation Framework for Benchmarking Entity Recognition and Disambiguation Systems

Hui Chen[*], Baogang Wei[†], Yiming Li[1], Yonghuai Liu[2], and Wenhao Zhu[3]

[1]College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, P.R. China
[2]Department of Computer Science, Aberystwyth University, Ceredigion SY23 3DB, UK
[3]College of Computer Engineering and Science, Shanghai University, Shanghai 200000, P.R. China

Abstract

Entity recognition and disambiguation (ERD) is a crucial technique for knowledge base population and information extraction. In recent years, numerous papers have been published on this subject, and various ERD systems have been developed. However, there is still some confusion over the ERD field for a fair and complete comparison of these systems. Therefore, it is of emerging interest to develop a unified evaluation framework. This paper presents an easy-to-use evaluation framework (EUEF), which aims at facilitating the evaluation process, and giving a fair comparison of ERD systems. EUEF is well designed and released to the public as open source, and thus could be easily extended with novel ERD systems, datasets, and evaluation metrics. It is easy to discover the advantages and disadvantages of a specific ERD system and its components based on EUEF. We perform a comparison of several popular and publicly available ERD systems by EUEF, and draw some interesting conclusions after a detailed analysis.

1 Introduction

Entity recognition and disambiguation (ERD) is a crucial technique to discover knowledge in texts, which would facilitate different tasks such as information extraction (IE), knowledge base population (KBP), and natural language processing (NLP). Generally, there are two variants of the ERD task: Wikification and Named Entity Linking (NEL), and we use ERD to refer to both of them in this paper. Recent literature has introduced a variety of ERD systems. However, there is still some confusion over the performances of these ERD systems, because they are generally evaluated using different datasets and evaluation metrics.

(Ling et al., 2015) argues the confusion in three aspects: (i) There is no standard definition of the ERD task; (ii) ERD systems are rarely compared using the same datasets and evaluation metrics; (iii) There is a lack of understanding of which aspect of a system is better than another. These problems have given rise to the development of a framework to unify and facilitate the evaluation process. Therefore this paper proposes a flexible and easy-to-use evaluation framework (EUEF). EUEF defines a series of matching and evaluation metrics which ensure a fair comparison among different ERD systems. EUEF also helps to improve an ERD system by discovering the strengths and weaknesses of its components. ERD task usually has a referential knowledge base (KB) that contains many entities as disambiguation targets. Most previous systems adopt Wikipedia, as it not only has abundant structural information, but also includes massive unstructured text information. Considering this scenario, as well as keeping consistency with previous work, EUEF also adopts the current version of Wikipedia as the referential KB.

The development of an evaluation framework for ERD systems has been mentioned in a few previous papers (Cornolti et al., 2013; Usbeck et al., 2015). EUEF is similar to them in some respects, but goes beyond them in several dimensions: (i) EUEF puts forward new matching metrics that are different from previous work; (ii) EUEF adopts a new method to process and evaluate **NIL**s (NIL is defined in Section 3), while previous frameworks usually overlook them; (iii) EUEF evaluates and analyzes the components of an ERD system more concretely, and the architecture of EUEF is refined and well designed, which is easily-extensible and

[*]chenhuicn@126.com
[†]wbg@zju.edu.cn

easy to use.

In this paper, we introduce a flexible and easy-to-use evaluation framework for benchmarking ERD systems, which is open source[1] and has good extendibility. EUEF has already integrated several popular publicly available ERD systems, datasets, and evaluation metrics. The motivation of this work is to make an attempt to facilitate and unify the evaluation process of ERD systems, as well as present a framework for analyzing the advantages and disadvantages of a specific ERD system. Our contributions are mainly in three aspects: (i) We propose an evaluation framework EUEF for ERD systems and make it publicly available; (ii) We propose several new matching metrics as well as a new approach to process and evaluate NILs; (iii) Based on the analysis of various ERD systems' performance, we give some suggestions for designing a better ERD system.

The rest of this paper is organized as follows: Section 2 describes some related work. Section 3 introduces some terminologies used in this paper. EUEF is detailed in Section 4. Experiments and discussions are presented Section 5. Finally, Section 6 draws some conclusions and indicates some future work.

2 Related Work

A variety of ERD systems have been proposed so far, ranging from pipeline and joint inference models to deep neural networks, and (Shen et al., 2015) give a survey about the ERD task. However, it is still difficult to understand the state of the art for ERD, as previous proposed approaches are usually evaluated with non comparable evaluation metrics over different datasets. This variation necessitates the development of a unified evaluation framework.

The BAT framework (Cornolti et al., 2013) is the first proposed framework designed for a fair comparison of various ERD systems, to the best of our knowledge. This framework defines a set of tasks as well as matching and evaluation metrics. It evaluates seven ERD systems on five datasets by using Wikipedia as the referential knowledge base. However, the matching metrics defined in the BAT framework is restrictive. (Rizzo et al., 2014) evaluate a bundle of ERD systems, and combine them using a machine learning algorithm to form a new ERD system. The proposed system chooses DBpedia (Bizer et al., 2009) as a referential knowledge base, and makes a mapping between Wikipedia and DBpedia. However, this work is inclined to developing a new ERD system rather than an evaluation framework. (Hachey et al., 2014) have designed an evaluation tool based on the AIDA-YAGO dataset (Hoffart et al., 2011), which extends the BAT framework by adding an isolated evaluation of disambiguation. But this framework does not evaluate mentions and NILs. (Usbeck et al., 2015) propose an evaluation framework GERBIL by extending the BAT framework. GERBIL provides a web service API[2] and allows access to the platform through some permanent URLs and NIF-based parameter data.

Compared with these previous works, EUEF has several advantages. EUEF not only assesses the comprehensive performance of an ERD system but also evaluates the components, which provides a better analysis of the system. We could discover the strengths and weaknesses of an ERD system based on the component evaluations. In addition, EUEF evaluates NILs, which are usually overlooked in previous frameworks. Finally, EUEF is designed for well extendibility and we make it publicly available as open source.

3 Terminologies

As mentioned in Section 1, EUEF adopts Wikipedia as the referential knowledge base, so all Wikipedia titles (articles) are the referential target entities, denoted by T.

- A document d is a plain text file.
- A confidence score is a real number, denoted by s and $s \in [0, 1]$.
- A mention is a phrase embedded in d, which is used to refer to something in the real world. Each mention is denoted by m and m is a triple $< p, l, s >$, where p is the position of the occurrence of the mention in d, l is the length of the mention, and s is the confidence in recognizing the mention.
- An entity is an instance of T, denoted by e and $e \in T$.
- A $null$ is a label parallel to an entity, representing all referential targets that are not contained in T. Thus $\{null\} \cap T = \emptyset$ and

[1] https://github.com/htlchh/EUEF

[2] http://aksw.org/Projects/GERBIL.html

$\{null\} \cup T$ is the universal set of referential targets.

- An annotation is a triple representing the mapping of a mention to an entity with a confidence, which is denoted by a and a is $< m, e, s >$.
- A NIL is also a triple $< m, null, s >$, which represents that the referential target of the mention does not exist in T, and s is the confidence in mapping m to $null$. NIL is a special annotation essentially.
- A candidate c consists of two parts: a mention m and a set of pairs: formally, $< m, \{< e_1, s_1 >, ...\} >$, and each pair is in the form of $< e, s >$, where e is a valid entity of the mention and s is the corresponding confidence. If the mention does not have any corresponding entities, then c is simplified as $< m, \{< null, s >\} >$.
- A dereference function df illustrates a many-to-one relation (Cornolti et al., 2013). Considering that Wikipedia has redirects, it is necessary to use df to normalize them to non-redirects when making comparisons. Formally, given two entities $e_1 \in T$ and $e_2 \in T$, $df(e_1)$ and $df(e_2)$ are two non-redirects, which are equal if and only if $df(e_1) = df(e_2)$.

Given a document, an ERD system would recognize a set of mentions, and create candidates for mentions, and finally generate a set of annotations and NILs.

4 The proposed framework

4.1 Architecture Overview

The architecture of EUEF is very concise, which is depicted in Fig. 1. Given an ERD system and a dataset, then picking a matching metric, EUEF would output the results after running the executor. The ERD systems, datasets, and matching metrics could be extended by implementing predefined interfaces. EUEF has already integrated several popular ERD systems, datasets, and evaluation metrics, which would be described in the following sections. However, the main intention of this paper is to introduce the evaluation framework, rather than make a comparison of all available ERD systems and datasets, while more ERD systems and datasets would be incorporated in future work.

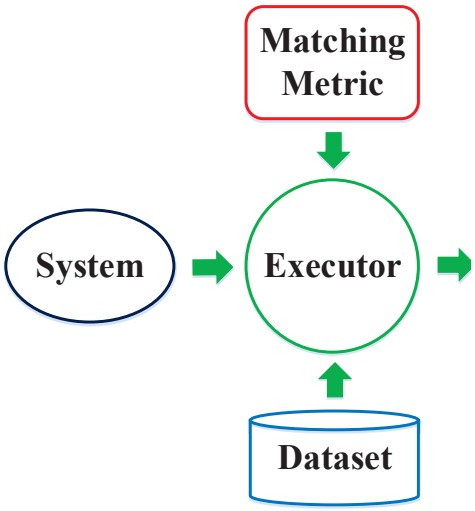


Figure 1: Architecture of EUEF, which is modular in design and well extensible.

4.2 Integrated ERD Systems

To this end, EUEF has integrated three ERD systems that are publicly available without any license keys or version issues. This section makes a brief description about these ERD systems.

- **Wikipedia Miner**: Wikipedia Miner is one of the earliest ERD systems and has been described in (David and H, 2008; Milne and Witten, 2013). This system starts by gathering all n-grams. Then it uses machine learning algorithms for mention recognition and disambiguation. Its mention recognition component is based on its disambiguation component. Wikipedia Miner annotates named entities as well as common concepts, but it does not produce NILs. The authors provide the source code and also a publicly available web service API.[3]
- **Illinois Wikifier**: Illinois Wikifier uses a local and global paradigm to solve the ERD

[3] http://wikipedia-miner.cms.waikato.ac.nz

problem, which is illustrated in (Ratinov et al., 2011). This system adopts the Illinois Named Entity Recognition (NER) (Ratinov and Roth, 2009) tool[4] to recognize mentions and does some postprocessing by predefined regular expressions. Then it treats the disambiguation process as a quadratic optimization problem. Wikifier only annotates named entities, and it does not produce NILs. The executable code of the system has been made publicly available, so it could be downloaded and run locally. The authors provide the software online.[5]

- **Priorer**: Priorer is a simple pipeline ERD system. Priorer applies the Stanford NER tool (Finkel et al., 2005) to recognize mentions, and then retrieves CrossWikis (Spitkovsky and Chang, 2012) to generate candidates by using a search engine.[6] CrossWikis is a dictionary that is created by crawling Wikipedia and Google cache and could be downloaded online.[7] Since each line of the CrossWikis is a mention with its possible entity associated with a confidence score, Priorer simply chooses the entity with the largest confidence as disambiguation target. If mentions have no candidates, or if the generated candidates are invalid (without corresponding Wikipedia titles), Priorer would transform these mentions into NILs. Priorer only annotates named entities, but it predicts NILs.

4.3 Integrated Datasets

Several publicly available datasets are collected from (Cornolti et al., 2013) and (Ratinov et al., 2011), which have already been introduced in previous work. Datasets such as AQUAINT, AIDA/CoNLL, IITB, MSNBC and ACE2004 are all appropriate for testing ERD systems, and EUEF has already integrated all of these datasets. If more datasets are available, it is also convenient to integrate them into EUEF. Basic statistical information about these datasets is shown in Table 1. Four of the integrated datasets are from newswire texts, and IITB consists of crawled web pages. Datasets IITB and AQUAINT contain gold standards consisting of named entities as well as common concepts. However, the other three datasets contain only named entities. This distinction of gold standards would make a significant impact on the performance of the component for mention recognition, which would be demonstrated in Section 5.1. The columns *Men*, *Ent*, and Ent_{dist} illustrate the multiplicity of mentions and entities of the dataset. EUEF has filtered those documents without any gold standards, as predictions for these documents would be all false positives, which would reduce the precisions. Some datasets contain embedded gold mentions, and EUEF filters out all these embedded mentions but the one with largest length when making an evaluation.

MSNBC, AIDA/CoNLL, and ACE2004 annotate NIL explicitly, while the other two datasets do not. EUEF classifies NIL into two groups: explicit NIL and implicit NIL. Explicit NIL is the annotation that is labeled with *null* (*none* or *nme*, etc.). For example, in MSNBC, a mention *Dana* is assigned with a *null* label, therefore ¡$Dana, null, 1$¿ is an explicit NIL. Implicit NIL is the annotation whose entity is already invalid according to Wikipedia. For instance, a mention *Al Goldman* is annotated with the entity *Al Goldman* in MSNBC. However, the article about *Al Goldman* does not exist in Wikipedia and this annotation is already deprecated. These deprecated annotations are treated as NILs as well. EUEF introduces implicit NIL for two reasons: (i) The deprecated annotations are out of work mainly because the corresponding entities are invalid. However, the mentions are still gold standards, and it is natural to transform these annotations into NILs. (ii) If no preprocessing is conducted of these deprecated annotations, whatever an ERD system predicts for the mentions, it will always result in false positives.

4.4 Integrated Matching Metrics

A matching metric is a Boolean function for comparing the results generated by ERD systems and gold standards. Each matching metric defines some constraints; the results are correct if and only if they satisfy the defined constraints compared to gold standards. Differing from previous frameworks, EUEF defines only one type of matching metric: fuzzy matching metric, which covers the

[4] http://cogcomp.cs.illinois.edu/page/software_view/NETagger

[5] http://cogcomp.cs.illinois.edu/page/software/

[6] http://lucene.apache.org

[7] http://nlp.stanford.edu/pubs/crosswikis-data.tar.bz2/

Table 1: Statistical information about datasets

Dataset	T_{doc}	T_{gold}	Doc	Men	Ent	Ent_{dist}	Ann	NIL	AVG_{doc}
AIDA/TestA	News	NE	215	5904	2991	1643	4787	1117	1243
AIDA/TestB	News	NE	231	5616	2924	1539	4485	1131	1040
AIDA/Training	News	NE	946	23396	11942	4087	18539	4857	1126
MSNBC	News	NE	20	755	340	290	658	97	3316
IITB	Web Pages	Mixed	649	18308	6566	3740	11085	7223	7191
ACE2004	News	NE	36	306	273	183	253	53	2258
AQUAINT	News	Mixed	50	727	727	573	727	0	1416

T_{doc} is the type of source documents in the dataset. T_{gold} is the type of gold standards, and *NE* means that only named entities are annotated, while *Mixed* means common concepts are also annotated. *Doc* is the total number of documents in the dataset. *Men* is the total number of mentions. *Ent* is the total number of entities. Ent_{dist} is the total number of distinct entities in the dataset. *Ann* is the total number of annotations. *NIL* is the total number of NILs. AVG_{doc} is the average length per document in the dataset.

strong matching metric and *weak matching metric* defined in (Cornolti et al., 2013). The confidence score associated with mention (candidate, annotation, and NIL) is used only for ranking and filtering in recognition and disambiguation phases, but is not used in the matching phase. Thus, the confidences would be discarded by choosing a best threshold when making an evaluation.

4.4.1 Mention Matching Metric

To define a fine matching metric between two mentions is challenging, because it involves two dimensions: the syntactic one and the semantic one (Cornolti et al., 2013). EUEF implements only syntactic matchings so far, and would consider semantic matchings in the future. It seems inappropriate if only comparing two mentions in accordance to their exact syntactics, as human annotators would annotate mentions with their preferences and bring bias in gold standard mentions. We have sampled documents from MSNBC and marked the mentions manually. Compared with the original labels, the Kappa coefficient (Carletta, 1996) indicates an agreement ratio with a score of 0.69. For example, *Home Depot Inc* and *Wal-Mart Stores Inc.* are two gold mentions about companies, but they are annotated with an inconsistent style as one ends with a period while the other does not. If an ERD system makes predictions such as *Home Depot Inc.* and *Wal-Mart Stores Inc*, the results are both false positives for missing or containing a period, and this is not expected. To tackle with problem, EUEF introduces a fuzzy matching metric based on edit distance (Ristad and Yianilos, 1998). First, we define two functions: *equal*(m_1,m_2) and *overlap*(m_1,m_2) as

$$equal(m_1, m_2) = \begin{cases} 1 & p_1 = p2 \wedge l_1 = l_2, \\ 0 & \text{else.} \end{cases}$$

and

$$overlap(m_1, m_2) = \begin{cases} 1 & (p_1 \leq p_2 \wedge p_2 \leq (p_1 + l_1)) \vee \\ & (p_2 \leq p_1 \wedge p_1 \leq (p_2 + l_2)) \vee \\ & (p_2 \leq p_1 \wedge (p_1 + l_1) \leq (p_2 + l_2)) \vee \\ & (p_1 \leq p_2 \wedge (p_2 + l_2) \leq (p_1 + l_1)), \\ 0 & \text{else.} \end{cases}$$

where m_1 and m_2 are two given mentions. The function *equal* tests whether two mentions are exactly syntactically matched, while the function *overlap* measures whether two mentions are overlapped. The *equal* and *overlap* functions are used in (Cornolti et al., 2013; Usbeck et al., 2015) as the cores of strong matching and weak matching, respectively.

Let M denote a set of mentions generated by an ERD system and G denote gold standard mentions, and then the fuzzy mention matching metric *MM* is defined as

$$\mathit{MM}(m, m') = \begin{cases} 1 & ned(m, m') \geq t, \\ 0 & \text{else.} \end{cases}$$

where $m \in M$, $m' \in G$, and $t \in [0, 1]$ is a given threshold. The function $ned(m, m')$ represents normalized edit distance, which is defined as:

$$ned(m, m') = \frac{ed(m, m')}{max(|m|, |m'|)}$$

where $ed(m, m')$ is the edit distance of m and m' and $|\bullet|$ represents the length of a string. If choosing a threshold $t = 1$, then *MM* is exactly the $equal$ function, while with a threshold $t = 0$, *MM* is equivalent to the $overlap$ function. Hence the two mention matching metrics defined in (Cornolti et al., 2013) and (Usbeck et al., 2015) could be deduced to *MM*. Some ERD systems would generate embedded mentions, for example, the mention

New York is embedded in the mention *New York Stock Exchange*. EUEF adopts a co-reference step for preprocessing the embedded mentions. If two or more mentions, which are generated by an ERD system, are embedded, EUEF would choose the one with the largest string length and discard the others, as it is considered that the long mention would be more representative and less ambiguous. Otherwise, if two or more embedded mentions are generated without co-reference, they may lead to more than one true positive according to *MM* when evaluated.

4.4.2 Candidate Matching Metric

Candidate matching metric *CM* is used to evaluate the performance of the component taht generates candidates. *CM* is based on the *MM*, as a mention is embedded in a candidate. Let C denote a set of candidates generated by an ERD system, and G denote gold standard candidates; then *CM* is defined as

$$CM(c, c') = \begin{cases} 1 & MM(m, m') = 1 \wedge ps \cap ps' \neq \emptyset, \\ 0 & \text{else.} \end{cases}$$

where $c \in C$, $c' \in G$, m is the embedded mention of c, m' is the embedded gold mention of c', ps is the targets entity-score pair set of m, and ps' is the gold entity-score pair set of m'. Since ps and ps' contain a series of entities, ps and ps' should be dereferenced by the df function first. Two candidates are matched if and only if their mentions are matched according to the given *MM*, and then there is at least one common referential entity (or *null*) for the two mentions. EUEF also does a co-reference preprocessing step to identify candidates whose mentions are embedded.

4.4.3 Disambiguation Matching Metric

In the entity disambiguation task, the gold mentions are given along with documents, and the only thing to do is to find the target entities of the given mentions. However, EUEF does not define a disambiguation metric explicitly for two main reasons: First, for an ERD system, the performance of the disambiguation component could be estimated from *MM* and *AM* (Section 4.4.4). Namely, for acquiring the performance of *MM*, it is easy to get the total number of correctly recognized mentions. For acquiring the performance of *AM*, it is also easy to get the total number of correctly disambiguated mentions. Then the performance of disambiguation could be estimated by a simple division of these two numbers. Since the integrated systems' disambiguation algorithms are not available, EUEF evaluates the performance of the involved ERD systems by this means. Second, if gold mentions and the disambiguation algorithm are both available, it is also easy to transform these gold mentions and their corresponding generated disambiguation targets into annotations, and then adopt *AM* to evaluate the results.

4.4.4 Annotation Matching Metric

Annotation matching metric *AM* is an end-to-end evaluation of an ERD system. An annotation is a triple ¡m, e, s¿, and thus the matching consists of two parts: mention matching metric *MM* and entity matching metric *EM*. *MM* has already been defined above. If an ERD system captures a set of entities E for a document, and let G denote the gold standard entities, *EM* is defined as

$$EM(e, e') = \begin{cases} 1 & df(e) = df(e') \\ 0 & \text{else.} \end{cases}$$

where $e \in E$ and $e' \in G$. *EM* exactly measures the matching of entities that are already dereferenced. Based on *MM* and *EM*, the annotation matching metric AM could be defined. Let A denote a set of annotations generated by an ERD system, and G denote gold standard annotations; then AM is

$$AM(a, a') = \begin{cases} 1 & MM(m, m') = 1 \wedge EM(e, e') = 1, \\ 0 & \text{else.} \end{cases}$$

where $a \in A$, $a' \in G$, e is the disambiguated entity of a, and e' is the gold entity of a'. EUEF does a co-reference preprocessing phase and dereferences the entities before comparison.

4.4.5 NIL Matching Metric

A NIL is a special annotation essentially by replacing the entity with a *null* label. Therefore the NIL matching metric *NM* is similar to *AM*. Let N denote a set of NILs generated by an ERD system, and G denote gold standard NILs; then *NM* is defined as

$$NM(n, n') = \begin{cases} 1 & MM(m, m') = 1 \\ 0 & \text{else.} \end{cases}$$

where $n \in N$, $n' \in G$, m is the embedded mention of n, and m' is the embedded gold mention of n'. The *null* labels are left out when comparing NILs, because they are always matched. Since NILs may contain embedded mentions, EUEF also does a co-reference preprocessing. The gold standard NILs are preprocessed as described in Section 4.3.

4.4.6 Matching Metrics for Deduced Tasks

The BAT framework (Cornolti et al., 2013) defines a set of annotation tasks, namely D2W, A2W, Sa2W, C2W, Sc2W, and Rc2W, and **suggests** that all the other tasks can be deduced from the Sa2W task. An ERD system usually generates annotations associated with confidence scores, which is exactly equal to the Sa2W task. Therefore, it is easy to evaluate these reduced tasks according to the reduction rules from the results output by ERD systems. It is noted that the C2W, Rc2W, and Sc2W tasks do not rely on mentions, and could be evaluated by using EM. The A2W task is similar to the Sa2W task but without associated confidence scores. Since confidence scores play no role in matching, it is natural to use AM to evaluate the A2W task. The D2W task is just the disambiguation task and has been discussed in Section 4.4.3. The Sa2W task could be evaluated by AM. The BAT framework does not define matching metrics for NILs. However, we could use *NM* to evaluate the given ERD system if it predicts NILs.

4.5 Evaluation Metrics

EUEF adopts two groups of the classical F1 measures: the macro group and the micro group. Let D denote a dataset and $d \in D$ a document, then precision and recall are defined as

$$P_{mic} = \frac{\sum_{d \in D} |TP_d|}{\sum_{d' \in D} |TP_{d'}| + |FP_{d'}|}$$

$$R_{mic} = \frac{\sum_{d \in D} |TP_d|}{\sum_{d' \in D} |TP_{d'}| + |FN_{d'}|}$$

$$P_{mac} = \frac{1}{|D|} \sum_{d \in D} \frac{|TP_d|}{|TP_d| + |FP_d|}$$

$$R_{mac} = \frac{1}{|D|} \sum_{d \in D} \frac{|TP_d|}{|TP_d| + |FN_d|}$$

where TP_d is the count of true positives of document d; FP_d is the count of false positives; and FN_d is the count of false negatives. Then, the F1 scores are defined as

$$F1_{mic} = \frac{2 * P_{mic} * R_{mic}}{P_{mic} + R_{mic}}$$

$$F1_{mac} = \frac{2 * P_{mac} * R_{mac}}{P_{mac} + R_{mac}}$$

In TAC-KBP (Ji et al., 2014; Ji et al., 2015), NIL is evaluated using clustering metrics that aim to discover new NIL clusters for populating knowledge bases. However, EUEF aims to evaluate the performance of the component for NIL generation and does not produce NIL clusters. Hence, EUEF also evaluates NIL by F1 measures for consistency.

5 Evaluation and Discussion

Robust experiments are conducted for the three ERD systems as they are without any training or tuning, and then the components of mention recognition, candidate generation, disambiguation, and the end-to-end system evaluation are performed by means of the matching metrics and evaluation metrics defined above. We chose two typical datasets MSNBC and AQUAINT, one of which only contains named entities while the other one includes common concepts as well, as illustration examples for performance comparison and analysis.

5.1 Mention Evaluation and Discussion

The performance of the mention recognition component is measured with the defined *MM*. In our evaluation example, the threshold t of edit distance is simply assigned to 0, and *MM* is the same as the weak matching introduced in (Cornolti et al., 2013). Wikipedia Miner and Wikifier produce the confidence scores associated with mentions. However, Priorer does not output confidence scores, and EUEF sets the default score as 1. Table 2 shows the results of the three ERD systems evaluated over two datasets. Priorer achieves the best performance over MSNBC in terms of all precisions, recalls, and F1 scores. Wikifier performs a little better than Wikipedia Miner in precisions and F1 scores, but with close recalls. As for AQUAINT, even though Priorer achieves the best precisions and F1 scores, the recalls are relatively poor. Compared with Priorer, Wikifier performs a little worse in precisions and F1 scores but comparable in recalls, and Wikipedia Miner achieves the best recalls.

Two examples are selected to illustrate the performance of *MM*. First, consider two gold mentions *Home Depot Inc* and *Wal-Mart Stores Inc.*, the latter is annotated by ending with a period

Table 2: Evaluation results based on MM

Dataset	System	P_{mic}	R_{mic}	$F1_{mic}$	P_{mac}	R_{mac}	$F1_{mac}$	TP	FP	FN
MSNBC	Wikpedia Miner	0.289	0.717	0.412	0.293	0.747	0.420	534	1313	211
	Wikifier	0.407	0.719	0.520	0.400	0.733	0.517	536	780	209
	Priorer	**0.896**	**0.905**	**0.900**	**0.877**	**0.885**	**0.881**	674	78	71
AQUAINT	Wikipedia Miner	0.289	**0.933**	0.442	0.289	**0.933**	0.441	678	1665	49
	Wikifier	0.280	0.582	0.378	0.276	0.572	0.372	423	1088	304
	Priorer	**0.407**	0.568	**0.474**	**0.410**	0.556	**0.472**	413	602	314

P_{mic} is the micro-precision and R_{mic} is the micro-recall. $F1_{mic}$ indicates the micro-F1 score. P_{mac} is the macro-precision and R_{mac} is the macro-recall. $F1_{mac}$ indicates the macro-F1 score. TP is the total number of the true positives. FP means the total number of the false positives. FN is the total number of the false negatives. The entries in boldface represent the best micro and macro precisions, recalls and F1 scores.

while the other does not. However, an ERD system, for example Priorer, predicts two mentions *Home Depot Inc.* and *Wal-Mart Stores Inc*, and both of them are wrong if evaluated with exact syntactic matching, which is not satisfactory. However, this problem could be solved well by evaluating with *MM*. Then considering another gold mention *Institute for Supply Management*, and Priorer makes a prediction *Institute for Supply Management and*, which is obviously a false positive by using *MM*. These false positives could be filtered out by tuning the threshold of the edit distance.

As mentioned above, MSNBC annotates named entities, while AQUAINT annotates named entities as well as common concepts. Priorer and Wikifier perform better than Wikipedia Miner in MSNBC in spite of the fact that they make much fewer predictions, which indicates that if a dataset is annotated only with named entities, it is better to choose NER for mention recognition. However, if more common concepts are annotated in a dataset, e.g., AQUAINT, n-gram and dictionary-based methods could discover more mentions. There is one important distinction that NER works much better in discovering NILs than n-gram, as the latter would discard all mentions that are out of the dictionary.

5.2 Candidate Evaluation and Discussion

As Wikipedia Miner web service API returns annotations that consist of one mention and one corresponding entity, the evaluation based on *CM* is similar to that with *AM*. Wikifier and Priorer both generate a list of entities for each mention. Mentions could be disambiguated correctly if and only if their candidates capture the correct entities. Recall is more important than precision and F1 score in the candidate generation phase, as it illustrates the upper bound of mentions which would be disambiguated correctly. EUEF computes two types of recalls in Table 3: the recall R_g compared with the gold standards, and the recall R_m compared with the recognized mentions. Priorer achieves the highest scores of R_g and R_m over MSNBC. As for AQUAINT, Wikipedia Miner gets the highest R_g score, while Priorer produces the highest R_m score.

Table 3: Evaluation results based on CM

Dataset	System	R_g	R_m	TP	FN
MSNBC	Miner	0.499	0.697	372	373
	Wikifier	0.528	0.733	393	352
	Priorer	**0.816**	**0.902**	608	137
AQUAINT	Miner	**0.795**	0.853	578	149
	Wikifier	0.497	0.853	361	366
	Priorer	0.491	**0.864**	357	370

$Miner$ represents Wikipedia Miner. R_g indicates the recall compared with all gold standards. R_m is the recall compared with the recognized mentions. TP is the total number of true positives, and FN represents the total number of false negatives. The entries in boldface represent the best recalls on two datasets.

CM is based on *MM*, and therefore the count of TP derived from *MM* is an upper bound of *CM*. All these systems generate candidates by retrieving dictionaries. R_g is decided by the mention recognition component and the dictionary together, while R_m is only relevant to the dictionary. Wikipedia Miner and Wikifier both adopt Wikipedia's resources (anchors, titles, redirects) as the dictionary, while Priorer chooses CrossWikis. From Table 3, it can be seen that Priorer's R_m on both datasets is the highest, which shows that CrossWikis is a better dictionary for candidate generation than Wikipedia resources in these two datasets, and this conclusion is consistent with that of (Ling et al., 2015). Priorer produces a low R_g score in AQUAINT mainly due to its poor performance in the mention recognition.

5.3 Disambiguation Evaluation and Discussion

Disambiguating the given mentions according to Wikipedia is the popular D2W task. As mentioned in Section 4.4.3, EUEF does not define an explicit disambiguation matching metric, and the disambiguation performance is estimated from *MM* and *AM*. Since in the disambiguation phase the mentions are given, the precision, recall and F1 score are equivalent, and the accuracy is usually chosen instead as the evaluation metric. The results of disambiguation components are shown in Table 4. Wikipedia Miner achieves the best accuracies over both datasets. Wikifier performs better than Priorer in disambiguation.

Table 4: Evaluation results of disambiguation

Dataset	System	Accuracy	M_g	M_s
MSNBC	Wikipedia Miner	**0.700**	534	374
	Wikifier	0.644	536	345
	Priorer	0.540	674	364
AQUAINT	Wikipedia Miner	**0.850**	678	576
	Wikifier	0.790	423	334
	Priorer	0.727	413	296

Accuracy is the percentage of the correctly disambiguated mentions. M_g is the total number of gold mentions which an ERD system generates. M_s is the total number of mentions which are disambiguated correctly by an ERD system. The entries in boldface represent the best accuracies achieved by Wikipedia Miner on two datasets.

Wikipedia Miner's disambiguation component adopts a classifier tuned on three features: prior probability, context relatedness and quality. Wikifier disambiguates mentions as an optimization problem by combining local and global features. Priorer simply chooses the candidate with the maximum prior probability as the target entity. All systems obtain better scores over AQUAINT than over MSNBC, which implies that disambiguating common concepts is easier than disambiguating named entities. Priorer disambiguates mentions based on the prior probabilities of entities from CrossWikis, which is effective when the target entities have the maximum prior probabilities. However, it performs very poorly when disambiguating long-tailed entities. Wikipedia Miner and Wikifier both treat the disambiguation process as a learning to rank problem. Wikifier designs features based on the syntactic information and Wikipedia's linking structure, while Wikipedia Miner discards syntactic features with only three more semantic like features. Even though Wikipedia Miner's disambiguation algorithm is very concise, it still achieves the best performance on both datasets, which indicates that the disambiguation task relies more on semantic features rather than syntactic features.

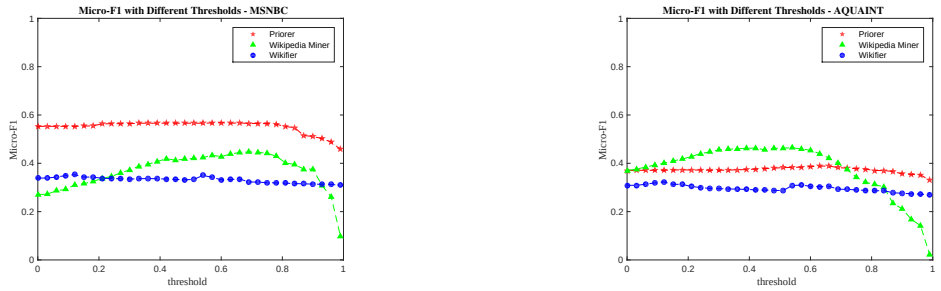
5.4 Annotation Evaluation and Discussion

Annotation evaluation is to test an ERD system's end-to-end performance. A system needs to balance its modules for the best comprehensive performance. Each system generates annotations with confidence scores; however, it is difficult to set the filtering threshold, which would play a vital role in the final performance measurement. To this end, we chose the best predictions of each system for evaluation by iterating the filtering threshold from 0 to 1 at intervals of a specific number. Fig. 2 shows the micro-F1 scores of the three systems over two datasets with multiple thresholds. As shown in Fig. 2(a), Priorer's performance over MSNBC is best with a micro-F1 score of 0.568, and Wikipedia Miner achieves a micro-F1 score of 0.447. Wikifier performs worst with a score of 0.366. Fig. 2(b) shows the results over dataset AQUAINT. Wikipedia Miner obtains the best micro-F1 score of 0.464, and Priorer's score is 0.389, while Wikifier achieves a score of only 0.323.

The main idea of Wikipedia Miner is to achieve high recall in the mention recognition phase with a relatively low but tolerable precision, and then refine the results in the following phases which mainly aim to improve precision. However, Wikifier and Priorer try to perform as best as possible at each step. As mentioned above, Priorer performs well in the mention recognition and candidate generation phases. Even though its disambiguation performance is the worst, it still works best over MSNBC and achieves a comparable result over AQUAINT, which illustrates that this simple method is a strong baseline for the ERD task. Wikipedia Miner works well in balancing the precision and recall, and finally achieves good comprehensive performance over both datasets. As shown in Fig. 2, the shape of lines in two subfigures is similar, which indicates the robust performance of these systems over different datasets. Wikifier's and Priorer's performance is stable as the threshold varies, but Wikipedia Miner is more sensitive to the threshold value.

5.5 NIL Evaluation and Discussion

Even though n-gram and dictionary-based methods would obtain high recalls, they have a vital drawback, that is, their capacity for recognizing unknown but potential entities is rather lim-



(a) MSNBC (b) AQUAINT

Figure 2: Results on three ERD systems' performance in two datasets with iterative thresholds.

ited, as these unknown entities would be discarded if they are out of the dictionary. However, approaches based on NER would achieve a better performance for recognizing unknown entities, especially named entities. Wikipedia Miner does not predict NILs for its restriction of its mention recognition component. Even though Wikifier adopts NER for mention recognition, it does not make a prediction for NILs. Priorer integrates a simple NIL component. If the recognized mentions have no candidates, or the generated candidates do not exist in Wikipedia, Priorer would annotate these mentions as NILs. As dataset AQUAINT has no explicit or implicit NILs, it is replaced with ACE for NIL evaluation, and the results are shown in Table 5.

Table 5: NIL evaluation results based on NM

System	Matching	Dataset	P_{mic}	R_{mic}	$F1_{mic}$
Priorer	*NM*	MSNBC	0.831	0.621	0.711
		ACE	0.350	0.749	0.478

P_{mic} indicates the micro-precision. R_{mic} indicates the micro-recall. $F1_{mic}$ indicates the micro-F1 score.

Priorer's performance over NILs is better than annotation over MSNBC, and the improvement is mainly due to no disambiguation step in the NIL evaluation task. The precision in ACE is not very high, as the dataset contains only annotations that are relevant to the main idea of the current document, that Priorer makes exhaustive predictions. However, the capacity of recognizing NIL is important for an ERD system, and would be in favor of finding novel potential entities. Priorer's NIL component is very simple and natural, while more effective methods would be investigated in future work.

5.6 Evaluation Summary

The integrated ERD systems and their components have already been evaluated comprehensively. Based on the analysis of their performance, we could draw several interesting conclusions: (i) NER methods are more appropriate for finding named entities, especially for predicting NILs, while ngram-based methods usually aim at achieving a high recall. (ii) It is interesting to note that good performance of a specific component of an ERD system may have limited contributions to the overall performance, and a system needs to make a trade-off between its components. (iii) It is useful to discover the advantages and disadvantages of different ERD systems, which would help design a better ERD system by combining the

well-working components. For example, improving the Priorer's disambiguation algorithm referring to other systems would endow it with a better comprehensive performance.

6 Conclusions

In this paper, we proposed an evaluation framework called EUEF for benchmarking ERD systems. EUEF aims at facilitating the evaluation process and giving fair comparison and detailed analysis of various ERD systems. EUEF is flexible and easy to use, which could be extended with novel ERD systems, datasets, and evaluation metrics conveniently. We make it publicly available as open source. EUEF has defined several new fuzzy matching metrics, and we proposed a new method to evaluate NILs. With fair and exhaustive comparisons based on EUEF, it is more convenient to discover the advantages and disadvantages of various ERD systems.

We also identified some shortcomings when developing EUEF, which would be resolved in future work. For example, for the mention matching metric it is crucial to combine the semantic information and the syntactic information together, while EUEF considers only syntactic information at present. We believe our framework would be helpful in the development of better quality ERD systems.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61572434), the China Knowledge Centre for Engineering Sciences and Technology (No. CKC-EST-2015-2-5), and the Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP) (20130101110-136).

References

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165.

Jean Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational linguistics*, 22(2):249–254.

Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. 2013. A framework for benchmarking entity-annotation systems. In *Proceedings of the 22nd international conference on World Wide Web*, pages 249–260. ACM.

Milne David and Witten Ian H. 2008. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics.

Ben Hachey, Joel Nothman, and Will Radford. 2014. Cheap and easy entity evaluation. 2:464–469.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792.

Heng Ji, HT Dang, J Nothman, and B Hachey. 2014. Overview of tac-kbp2014 entity discovery and linking tasks. In *Proc. Text Analysis Conference (TAC2014)*.

Heng Ji, Joel Nothman, and Ben Hachey. 2015. Overview of tac-kbp2015 tri-lingual entity discovery and linking. In *Proc. Text Analysis Conference (TAC2015)*.

Xiao Ling, Sameer Singh, and Daniel S Weld. 2015. Design challenges for entity linking. *Transactions of the Association for Computational Linguistics*, 3:315–328.

David Milne and Ian H Witten. 2013. An open-source toolkit for mining wikipedia. *Artificial Intelligence*, 194:222–239.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, pages 147–155. Association for Computational Linguistics.

Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. 2011. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384. Association for Computational Linguistics.

Eric Sven Ristad and Peter N Yianilos. 1998. Learning string-edit distance. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(5):522–532.

Giuseppe Rizzo, Marieke van Erp, and Raphaël Troncy. 2014. Benchmarking the extraction and disambiguation of named entities on the semantic web. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*.

Wei Shen, Jianyong Wang, and Jiawei Han. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *Knowledge and Data Engineering, IEEE Transactions on*, 27(2):443–460.

Valentin I Spitkovsky and Angel X Chang. 2012. A cross-lingual dictionary for english wikipedia concepts. In *LREC*, pages 3168–3175.

Ricardo Usbeck, Michael Röder, Axel-Cyrille Ngonga Ngomo, Ciro Baron, Andreas Both, Martin Brümmer, Diego Ceccarelli, Marco Cornolti, Didier Cherix, Bernd Eickmann, et al. 2015. Gerbil: General entity annotator benchmarking framework. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1133–1143. ACM.